A Bio-Inspired Robust Adaptive Random Search Algorithm for Distributed Beamforming

Chia-Shiang Tseng, Chang-Ching Chen, Che Lin
Institute of Communication Engineering & Department of Electrical Engineering
National Tsing Hua University, Hsinchu, Taiwan 30013
E-mail: s9964518, s9864511, clin@ee.nthu.edu.tw

Abstract—A bio-inspired robust adaptive random search algorithm (BioRARSA), designed for distributed beamforming for sensor and relay networks, is proposed in this work. It has been shown via a systematic framework that BioRARSA converges in probability and its convergence time scales linearly with the number of distributed transmitters. More importantly, extensive simulation results demonstrate that the proposed BioRARSA outperforms existing adaptive distributed beamforming schemes by as large as 29.8% on average. This increase in performance results from the fact that BioRARSA can adaptively adjust its sampling stepsize via the "swim" behavior inspired by the bacterial foraging mechanism. Hence, the convergence time of BioRARSA is insensitive to the initial sampling stepsize of the algorithm, which makes it robust against the dynamic nature of distributed wireless networks.

I. INTRODUCTION

In recent years, the advancement of microelectronic technology makes possible the manufacture of low-cost transceivers that can be deployed with ease to form distributed sensor and relay networks. Since then, studies in distributed network have attracted enormous attentions. However, there are innate limitations, e.g., the lack of constant power supply for distributed transmitters, that need to be particularly addressed in the applications of such networks. A viable solution to overcome this energy scarcity is for the distributed transmitters to form an antenna array and transmit collaboratively using distributed beamforming. The problem remains, however, as the beamforming coefficients for each distributed transmitters cannot be easily obtained without complete knowledge of the channel state information (CSI), which is in general not available at the transmitter end.

A simple yet efficient adaptive distributed beamforming scheme that requires only a single bit feedback was proposed by Mudumbai et al in [1]. The convergence analysis has been conducted using different approaches in the literature [1], [2], [3], [4], where the fast convergence and the linear scalability of the scheme have been well established. With these extensive analyses, it has been shown that the simple one-bit adaptive scheme provides exceptional performance. The challenge ahead is then to discover new schemes that can further improve the performance without sacrificing much on the complexity.

In [2], we reformulated this adaptive distributed beamforming scheme as a local random search algorithm under a systematic framework. Once reformulated in this random search framework, an immediately observation is that the performance can be improved by introducing *Reverse Tracking*[1], a rather old technique in the field of random search algorithms [5], [6]. Rigorous analysis has been carried out in [3] to show that the convergence rate can indeed be doubled once this intuitive idea is applied to the one-bit adaptive scheme. However, the exceptional performances of both the one-bit adaptive scheme and the improved reverse tracking scheme highly rely on proper tuning of system parameters with respect to the total number of transmitters, which may not be known a prior due to environmental variations or random node failures.

Surprisingly, the key to achieve breakthrough sometimes lies in a completely different field. In [7], a heuristic random search mechanism, that mimics the foraging behavior of E. Coli bacteria, was proposed. The randomized search strategy seems suitable for the phase alignment process of adaptive distributed beamforming, yet some features cannot be included due to the innate limitations of distributed networks. Therefore, we carefully redesign the bacterial foraging algorithm and propose a bio-inspired robust adaptive random search algorithm (BioRARSA) that can adaptively adjust its sampling stepsize through the "swim" behavior of E. coli bacteria. As a result, BioRARSA is robust against all initial parameters and adapts well to the dynamic nature of distributed networks.

We organize the paper as follows: In section II, the problem of adaptive distributed beamforming in distributed wireless networks is described, and its connection to a random search framework is established. The description of our proposed scheme, BioRARSA, is presented in section III. Furthermore, the convergence analysis using the analytical tools proposed in [4] is provided in section IV. In section V, we show numerical simulations to demonstrate some interesting characteristics of BioRARSA. Finally, we conclude the advantages of BioRARSA and touch upon future works in section VI.

II. PROBLEM FORMULATION

In this section, we introduce the problem of adaptive distributed beamforming. To systematically analyze such prob-

[1]Reverse tracking is based on the intuition that if moving in the forward direction decreases the objective function, it is highly possible that moving in the reverse direction improves the objective function.

lem, we further describe a recently proposed random search framework [4] that greatly facilitate the convergence analysis.

A. Distributed Beamforming with Limited Feedback

We begin with the formulation of the distributed beamforming problem. In our setting, N_s distributed transmitters seek to transmit a common message $s \in \mathbb{C}$, with power constraint $\mathrm{E}\left[|s|^2\right] \leq P$, to the remote receiver. Each transmitter and the receiver is equipped with a single antenna, where the channel between each transmitter and the receiver is assumed to be frequency flat and slow fading. Also, each transmitter is allowed to alter its beamforming coefficient to achieve phase alignment. We assume that the distributed transmitters can only cooperate via a one-bit, error-free reverse feedback link from the receiver, since local cooperation requires sophisticated media access control (MAC) layer protocols, which may result in large energy dissipation and is not desirable for low-complexity distributed networks.

Based on the system specification, the discrete-time, complex baseband model over one coherence interval is given by

$$y[n] = \sum_{i=1}^{N_s} h_i g_i[n] s + w[n] = \sum_{i=1}^{N_s} a_i b_i[n] e^{j(\phi_i + \psi_i[n])} s + w[n]$$

(1)

where $y[n] \in \mathbb{C}$ is the received signal, $h_i = a_i e^{j\phi_i} \in \mathbb{C}$ is the channel fading gain, $g_i[n] = b_i[n] e^{\psi_i[n]} \in \mathbb{C}$. is the beamforming coefficients, and $w[n] \sim \mathcal{CN}(0, \sigma^2)$ corresponds to the additive white Gaussian noise.

To simplify notations[2], we let $s = \sqrt{P}$ and impose fixed power constraint $b_i = 1$ among transmitters. Assume that the magnitude of the signal component can be perfectly estimated, the received signal magnitude can be expressed as

$$\mathrm{Mag}(\theta_1[n], \cdots, \theta_{N_s}[n]) = \sqrt{P} \left| \sum_{i=1}^{N_s} a_i e^{j\theta_i[n]} \right|$$

(2)

where $\theta_i[n] = \phi_i + \psi_i[n]$ is the total phase of the received signal from transmitter i at the n-th transmission.

In order to minimize detection error, it is clear we should maximize this magnitude function, and its global maxima are achieved when all phases are aligned, i.e., $\theta_1[n] = \theta_2[n] = \cdots = \theta_{N_s}[n]$. In this setting, distributed transmitters seek to achieve this phase alignment adaptively with the help of a low rate feedback link. Consequently, we term such problem as adaptive distributed beamforming.

B. A Random Search Framework

With no prior knowledge of the channel fading coefficients, the received signal magnitude function is unknown for our system. However, for the n-th transmission, the remote receiver obtains a sample of $\mathrm{Mag}(\cdot)$ at $\boldsymbol{\theta}[n] = [\theta_1[n] \ \cdots \ \theta_{N_s}[n]]$. As a result, we can model the adaptive distributed beamforming problem as follows.

Problem 1: Given an unknown function $f : \Theta \rightarrow \mathbb{R}$ where only samples of $f(\cdot)$ is available, determine the global maximum value of $f(\cdot)$.

For such problem, randomized schemes that utilize only samples of $\mathrm{Mag}(\cdot)$ seems to be viable solutions. To analyze the convergence behavior of such randomized schemes for Problem 1, we introduce the general random search framework proposed in [4]. The framework describes a general random search algorithm, which we restate as follows:

- **Step 0**: Initialize the algorithm by choosing an arbitrary initial point $\boldsymbol{\theta}[0] \in \Theta$.
- **Step 1**: Generate a random perturbation $\boldsymbol{\delta}[n]$ according to a probability measure μ_n, that could be time-varying.
- **Step 2**: Update $\boldsymbol{\theta}[n]$ by $\boldsymbol{\theta}[n] = D(\boldsymbol{\theta}[n-1], \boldsymbol{\delta}[n])$, where the mapping $D(\cdot, \cdot)$ satisfies the condition that $f(D(\boldsymbol{\theta}[n-1], \boldsymbol{\delta})) \geq f(\boldsymbol{\theta}[n-1])$.

Clearly, the study of random search algorithms is a much more mature field in the mathematical society [5], [6]. If the randomized scheme for adaptive distributed beamforming can be reformulated as the above random search algorithm, then the framework becomes a powerful tool that allows us to analyze the adaptive distributed beamforming schemes systematically.

The analysis of the one-bit scheme proposed by Mudumbai et al is an immediate example of the application of such framework as has been shown in [2], where Θ is equal to $\mathbb{R}^{N_s}$, and $f(\cdot)$ is the function $\mathrm{Mag}(\cdot)$. The probability measure in this case is uniformly distributed in a hypercube centered at the origin with sides equal to $2\Delta_0$. The function $D(\cdot, \cdot)$ is described by

$$D(\boldsymbol{\theta}[n-1], \boldsymbol{\delta}[n]) = \\ \boldsymbol{\theta}[n-1] + \boldsymbol{\delta}[n] \cdot 1_{\{\mathrm{Mag}(\boldsymbol{\theta}[n-1]+\boldsymbol{\delta}[n]) > \mathrm{Mag}(\boldsymbol{\theta}[n-1])\}}$$

(3)

where $1_{\{\cdot\}}$ is the indicator function. As a result, the one-bit scheme can be reformulated as a simple local random search with fixed sampling stepsize.

As has been shown in [1], [2], [4], the simple local random search algorithm performs surprisingly well. In this work, we hope to propose a new algorithm with even better performance, i.e, one that achieve phase alignment with less time or power, based on further understanding of the problem via the random search framework, and inspiration from the foraging behavior of E. coli bacteria in biology.

III. BIO-INSPIRED ROBUST ADAPTIVE RANDOM SEARCH ALGORITHM - BIORARSA

As indicated by its name, BioRARSA is inspired by the foraging behavior of E. coli bacteria. In [7], the foraging strategy and the life cycle of E. coli bacteria have been studied extensively. When foraging, a E. coli bacterium randomly chooses a direction and swims towards the chosen direction until the density of its food fails to improve. This randomized search strategy has been shown to be effective and applied in many fields. This motivates the inclusion of the "swim" behavior into our algorithm. By careful incorporation of the

---

[2]The assumptions aim only for notation simplicity. As has been reported in [4], random search framework can be used to handle systems with more complicated power constraints, e.g., individual/total power constraints.

Initialize parameters
repeat
    for $i \leftarrow 1$ to $L.Helds$ **do**
        $\boldsymbol{\delta}[k] \sim \text{uni}\left([-\Delta_k, \Delta_k]^{N_s}\right)$
        if $\text{Mag}(\boldsymbol{\theta}[k-1] + \boldsymbol{\delta}[k]) < \text{Mag}(\boldsymbol{\theta}[k-1])$ **then**
            $\boldsymbol{\delta}[k] \leftarrow -\boldsymbol{\delta}[k]$
        end if
        if $\text{Mag}(\boldsymbol{\theta}[k] + \boldsymbol{\delta}) > \text{Mag}(\boldsymbol{\theta}[k-1])$ **then**
            $\boldsymbol{\theta}[k] \leftarrow \boldsymbol{\theta}[k] + \boldsymbol{\delta}$
            $\beta_k \leftarrow 1$
            while $\text{Mag}(\boldsymbol{\theta}[k] + \boldsymbol{\delta}[k]) > \text{Mag}(\boldsymbol{\theta}[k])$
                and $\beta_k < L.Swim$ **do**
                $\boldsymbol{\theta}[k] \leftarrow \boldsymbol{\theta}[k] + \boldsymbol{\delta}[k]$
                $\beta_k \leftarrow \beta_k + 1$
            end while
        end if
        $k \leftarrow k + 1$
    end for
    $Avg.Swim \leftarrow \max\left(\alpha, \sum_{i=k-L.Held+1}^{k} \beta_k \;/\; L.Helds\right)$
    $\Delta_{k+1} \leftarrow \Delta_k \cdot Avg.Swim$
until Stopping criteria reached

---

foraging behavior into our algorithm, we propose BioRARSA that achieves superior performance and is highly robust against system uncertainty.

A. Algorithm Description

Now we introduce the proposed algorithm in details. The pseudocode of proposed algorithm is presented as Algorithm I in Table I, and we elaborate the algorithm step-by-step as follows.

Step 1 - Initialization: We first initializes the system by generating the initial phase $\boldsymbol{\theta}[0] \in \mathbb{R}^{N_s}$. Other parameters of the system, including $L.Held$, $L.Swim$, and Δ_0, can be transmitted from the receiver to the transmitters or pre-coded into the system in this stage. The purpose of these parameters will become clear in the illustration of later steps.

Step 2 - Random Perturbation: Assume now that the system is at the k-th iteration. Each distributed transmitter in the system selects a random perturbation $\delta_i[k]$ that is uniformly distributed within $[-\Delta_k, \Delta_k]$, where Δ_k is the sampling stepsize of the algorithm and can be adjusted iteratively. From the perspective of the entire system, a random vector $\boldsymbol{\delta}[k] = [\delta_1[k] \cdots \delta_{N_s}[k]]^T$ that is uniformly distributed in $[-\Delta_k, \Delta_k]^{N_s}$ is jointly generated by distributed transmitters.

Once the random perturbations are generated, distributed transmitters add the perturbations to their beamforming coefficients and transmit the common message s simultaneously. Upon receiving the composite signal $y[k]$, as in (1), the receiver compares the received signal magnitude with the largest magnitude recorded up to the $(k-1)$-th iteration, and broadcasts a one-bit feedback signal to the transmitters. The broadcasted binary signal is given by

$$S = \mathbb{1}_{\left\{\text{Mag}(\boldsymbol{\theta}') > \max_{i=1,\cdots,k-1} \text{Mag}(\boldsymbol{\theta}[i])\right\}} \tag{4}$$

where $\boldsymbol{\theta}' = \boldsymbol{\theta}[k-1] + \boldsymbol{\delta}[k]$. If "1" is broadcasted, i.e., received signal magnitude increased with perturbations, the algorithm enters **Step 4**. Otherwise, the algorithm enters **Step 3**.

Step 3 - Reverse tracking: Each distributed transmitter multiplies its random perturbation $\delta_i[k]$ generated in **Step 2** by -1, and use it to perturb the beamforming coefficients for the next transmission. After comparing the received signal magnitude with the previous ones, the receiver broadcasts a one-bit feedback signal back to transmitters as in **Step 2**. The broadcasted signal is the same as in (4), except that now $\boldsymbol{\theta}' = \boldsymbol{\theta}[k-1] - \boldsymbol{\delta}[k]$. If "1" is broadcasted, the algorithm enters **Step 4**. Otherwise, the algorithm enters **Step 5**.

Step 4 - Swim: As the perturbations generated in **Step 2**, or their reversed versions in **Step 3** increase the received signal magnitude, the algorithm will proceed to search along the same direction by adding the same perturbations to the beamforming coefficients. Again, the received signal magnitude achieved by these perturbed beamforming coefficients is compared with the previous ones as in **Step 2** and **Step 3**. If "1" is broadcasted, transmitters update the beamforming coefficients with the perturbed value and return to the beginning of **Step 4**. Otherwise, the algorithm enters **Step 5**. Note that this step mimics the swimming behavior of E. coli bacteria, since the algorithm will continue to move along a direction once it is found to be beneficial and until no further improvement can be made. As a result, we term **Step 4** as "swim" and the number of iterations spent in **Step 4** as "swim length".

In fact, **Steps 2 - 4** can be combined into a single expression given by

$$\boldsymbol{\theta}[k] = \boldsymbol{\theta}[k-1] + \beta_k \cdot \boldsymbol{\delta}[k] \tag{5}$$

where

$$\beta_k = \underset{N=0,\pm 1,\cdots,\pm L.Swim}{\arg\max} \text{Mag}(\boldsymbol{\theta}[k-1] + N\boldsymbol{\delta}[k]) \tag{6}$$

Notice that we impose a maximum swim length "$L.Swim$". This maximum swim length is set to prevent situations where the algorithm "swims" too far along a direction that only provides slight improvement on the performance. In this case, the expected improvement on the received signal magnitude through swimming is usually less than that achieved by generating new perturbations. It has been verified via simulations that imposing such limit improves overall performances.

Step 5 - Stepsize Adjustment: For every $L.Held$ iterations, the algorithm adjusts its sampling stepsize Δ_k according to the average swim length during the previous $L.Held$ iterations. Also, the adjustment made to Δ_k is no less than a factor α. In cases where the average swim length is close to zero, this α factor prevents the sampling stepsize Δ_k from becoming too small[3]. The stepsize adjustment can be described by

$$\Delta_k = \Delta_{k-1} \cdot \max\left(\alpha, \sum_{i=k-L.Held+1}^{k} |\beta_k| \;/\; L.Helds\right) \tag{7}$$

---

[3]Setting sampling stepsize with values that are too small can sometimes cause random search algorithms to virtually "trap" in the local region, and severely degrade algorithms' performance.

The overall effect can be seen as the algorithm adaptively increase the sampling stepsize if there were many "swims", which indicates that the perturbations generated were small on average, and decrease the sampling stepsize by a factor as small as α if there were only few or no "swims", which indicates that the search area is too large.

For iterations that involve no adjustment, the value of Δ_k remains unchanged as in the last iteration. No matter there is an adjustment or not, the algorithm returns to **Step 2**, which marks the end of the current iteration.

Stopping criterion: During each step of BioRARSA, if the received signal magnitude is greater than a target value[4], the algorithm exits and the phase alignment for transmit beamforming is said to be completed.

B. Important Characteristic of BioRARSA

As will become clear in section V, the convergence behavior of BioRARSA is either insensitive to the choice of parameters at the initialization or to the number of distributed transmitters in the system. Therefore, all necessary parameters can be pre-coded and transmission overheads in **Step 1** can be eliminated.

Notice that although the algorithm is more complicated than the one-bit adaptive scheme, there is neither extra requirement on the feedback information nor the need for information exchange between distributed transmitters. The added cost is the computational complexity to execute the algorithm in the microprocessor equipped in each transmitter. However, energy consumed for such computation is negligible compared with that consumed by transmission. BioRARSA is hence a solid improvement over existing schemes.

IV. CONVERGENCE ANALYSIS

The analysis of convergence can be greatly simplified via the use of the random search framework proposed in [4], which provides a general proof if the reformulated random search algorithm satisfies the following two conditions:

1) The origin is an interior point of the support of the probability measure μ_n for all n.
2) The objective function $f : \Theta \to \mathbb{R}$ is continuous and all of its local maxima are global maxima.

These properties jointly provide the sufficient conditions for a local random search algorithm to converge. Therefore, in the ensuing discussions, we reformulate BioRARSA as a random search algorithm and examine these sufficient conditions. Once these conditions are satisfied, we can then apply the framework to show the convergence and linear scalability of BioRARSA.

A. Reformulation and Sufficient Conditions

Based on (5) and (6), the reformulation is as follows:

$$f(\cdot) = \text{Mag}(\cdot) \tag{8}$$

$$\mu_n = \text{uni}([-\Delta_n, \Delta_n]^{N_s}) \tag{9}$$

$$\text{Range}\{D(\boldsymbol{\theta}, \boldsymbol{\delta}[n])\} = \{\boldsymbol{\theta} + \Omega\} \cap U_{\boldsymbol{\theta}} \tag{10}$$

[4]The target value can be based on statistics of the channel, which is assume to be available and known by receiver.

where

$$U_{\boldsymbol{\theta}} := \{\boldsymbol{\omega} \in \Theta \mid \text{Mag}(\boldsymbol{\omega}) \geq \text{Mag}(\boldsymbol{\theta})\}$$
$$\Omega = \{\hat{\boldsymbol{\omega}} | \hat{\boldsymbol{\omega}} = L.Swim \cdot \boldsymbol{\omega}, \boldsymbol{\omega} \in \text{supp}(\mu_n)\}$$

Unlike the original one-bit adaptive scheme, the support of $\boldsymbol{\delta}[n]$ is extended to Ω since the "swim" step allows the perturbation to reach further by a factor of $L.Swim$. Although the probability distribution in Ω is no longer uniform, it is clear that the origin still lies in the interior of Ω. Therefore, the first sufficient condition is satisfied.

The property of $\text{Mag}(\cdot)$ has been identified by Lin et al [2] that all its local maxima are global maxima. This property is important since it ensures that a local random search algorithm can never be trapped in local maxima. Hence, the second sufficient condition is also satisfied.

B. Proof of Convergence

As both sufficient conditions are satisfied, we proceed to the proof of convergence with the help of the framework in [4]. Since the framework guarantees convergence of the target algorithm once both sufficient conditions are satisfied, the proofs parallel those derived in [4] and are hence omitted here due to space constraint. The sufficient conditions lead to the following proposition.

Proporsition 1: For any given $\boldsymbol{\theta} \in \Theta \setminus R_\epsilon$ and $\Delta > 0$, there exists $\gamma > 0$ and $0 < \eta \leq 1$ such that

$$\Pr[f(\boldsymbol{\theta} + \boldsymbol{\delta}) - f(\boldsymbol{\theta}) \geq \gamma] \geq \eta$$

where $\boldsymbol{\delta}$ is a random vector generated with the probability measure μ_n, R_ϵ is the ϵ-convergence region as stated in [2], which is defined as $R_\epsilon := \{\boldsymbol{\theta} \in \Theta : f(\boldsymbol{\theta}) > f(\boldsymbol{\theta}^\star) - \epsilon\}$, and $\boldsymbol{\theta}^\star$ is a global maximum point.

Proposition 1 states that there is always a non-zero probability to improve the received signal magnitude within each iteration before the stopping criterion is satisfied. Based on this, the random search framework provides the following theorem

Theorem 1: The sequence $\{\boldsymbol{\theta}[n]\}_{n=1}^{\infty}$ converges in probability, i.e., given $\epsilon > 0$,

$$\lim_{n \to \infty} \Pr[\boldsymbol{\theta}[n] \in R_\epsilon] = 1$$

As the sequence converges to R_ϵ, the received signal magnitude becomes greater or equal to $\text{Mag}(\boldsymbol{\theta}^\star) - \epsilon$. By setting ϵ infinitely small, we prove the convergence of BioRARSA.

C. Linear Scalability

BioRARSA inherits the same desirable feature of linear scalability from its predecessor. This can also be proven by using the random search framework.

Theorem 2: The average time of convergence scales linearly with N_s. That is, there exists $k < \infty$, such that

$$\text{E}[\text{Mag}(\boldsymbol{\theta}[n])] \geq \text{Mag}(\boldsymbol{\theta}^\star) - \epsilon, \quad \forall n \geq k \cdot N_s.$$

The detailed proof is omitted due to space constraint, but a brief description is given here: From the proof in [2], it has
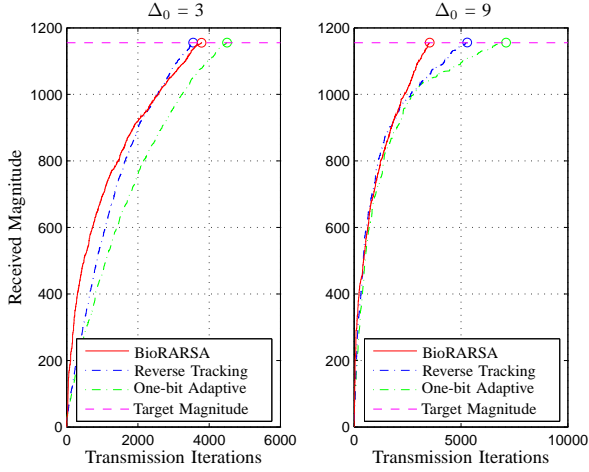
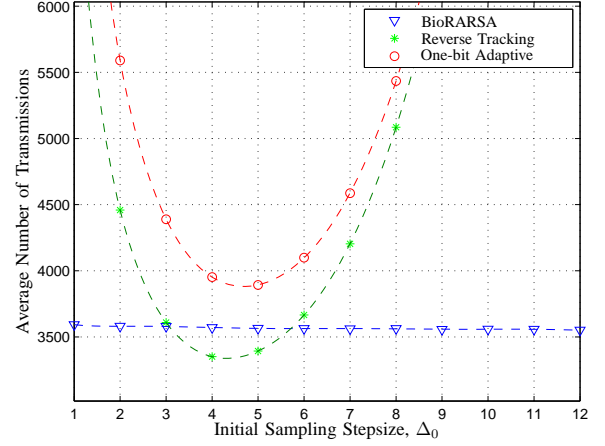Fig. 1. Trajectories of received signal magnitude of a 800-node system.

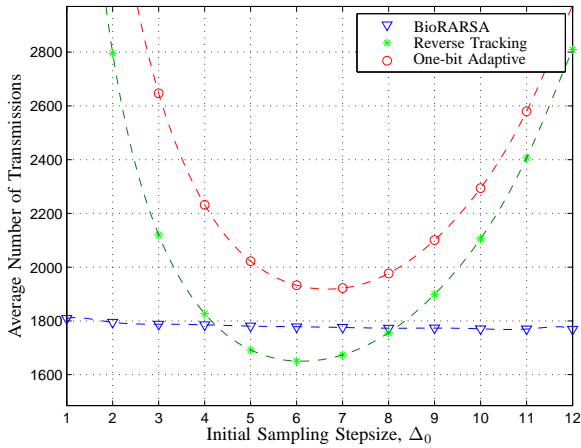Fig. 3. Performance comparison of 800-node systems.

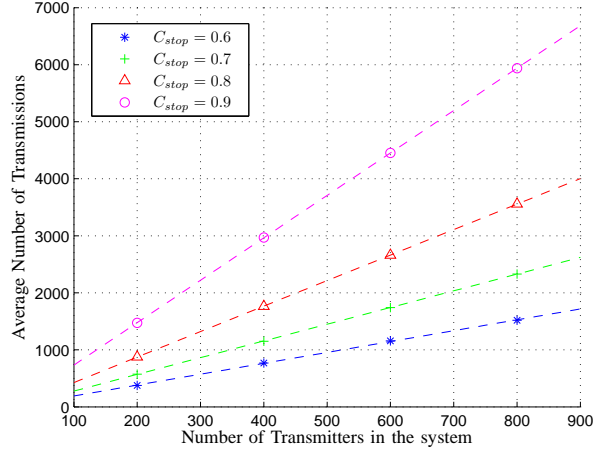Fig. 2. Performance comparison of 400-node systems.

Fig. 4. Linear scalability of BioRARSA.

been shown that the reformulated random search algorithm scales linearly with N_s. Also, notice that the total number of transmissions of BioRARSA is upper bounded by $L.Swim$ times the total number of transmissions of the reformulated algorithm described by (8)-(10). Since the reformulated algorithm is linearly scalable by the random search framework [2], BioRARSA must also be linearly scalable.

V. SIMULATION RESULT

Numerical results presented in this section are conducted using Monte Carlo simulations. Each point in Fig. 2 to 4 and values in Table II are averaged over 10^4 random instances that consist of 10^2 pre-generated channel realization $\{h_i\}$ each with 10^2 independent random perturbation sequences $\{\delta_i[n]\}$. Through simulations, we show the performance advantage and the robustness of BioRARSA, and proofs derived in Sec. IV are also validated.

The performance is measured by the average number of simultaneous transmissions from distributed transmitters to

the receiver before reaching the stopping criterion. We set $L.Helds$ and $L.Swim$ to 5 for these values are optimal for most systems. Simulation results that justify these choices of $L.Helds$ and $L.Swim$ are omitted due to space constraint.

To demonstrate the robustness of BioRARSA with respect to the initial choice of sampling stepsize Δ_0, we first present two sets of trajectories of received signal magnitude traversed by the randomized algorithms in Fig. 1. Each magnitude trajectory corresponds to a realization of the beamforming coefficients recorded during the phase alignment process using the stated schemes, and Δ_0 of each set is set to be $3°$ and $9°$, respectively. These trajectories confirm that BioRARSA is insensitive to the initial choice of Δ_0, while the number of transmissions in the reverse tracking scheme and the one-bit adaptive scheme increase nearly 60% when Δ_0 increase from $3°$ to $9°$. Note that we have generated these trajectories repeatedly for different realizations and obtained similar behaviors. For all realizations, all three algorithms converge without exception. This also validates our proof in Sec. IV.
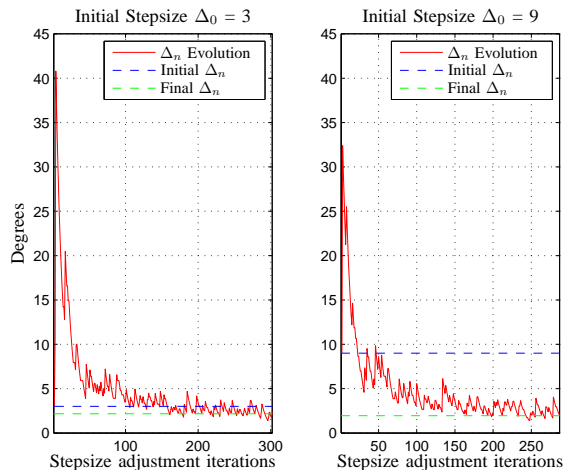
Fig. 5. Evolutions of sampling stepsizes in a 800-node system.

TABLE II
PERFORMANCE GAIN OF BIORARSA WITH RESPECT TO
THE REVERSE TRACKING SCHEME (IN %)

Δ_0 \ # Nodes	200	400	600	800
1°	72.3 %	63.5 %	57.1 %	51.9 %
3°	31.5 %	15.7 %	6.6 %	0.8 %
5°	6.2 %	-5.2 %	-7.4 %	-5.1 %
7°	-6.4 %	-6.1 %	3.4 %	15.3 %
9°	-8.7 %	6.6 %	26.8 %	44.9 %
11°	-3.1 %	26.5 %	52.5 %	70.9 %
Average Gain	15.3 %	16.8 %	23.1 %	29.8 %

Aside from its robust nature, the overall performance of BioRARSA is competitive compared with existing schemes. From Fig. 2 and 3, it can be identified that the one-bit adaptive scheme is completely outperformed by the reverse tracking scheme and BioRARSA, and there exists a small set of Δ_0 where reverse tracking outperforms BioRARSA if the reverse tracking scheme happens to choose the stepsizes that are close to the optimal values. However, the optimal choice of Δ_0 is closely related to the number of distributed transmitters in the system, which is often difficult to track in a distributed network. On the contrary, BioRARSA is capable of providing near optimal performance without any prior knowledge of the system, making it more suitable for distributed networks.

To provide a more detailed performance comparison between the reverse tracking scheme and BioRARSA, the normalized performance gain $(N_{RT} - N_{BioRARSA})/N_{RT}\cdot 100\%$ is presented in Table II, where N_{RT} and $N_{BioRARSA}$ are the average number of transmissions with respect to each scheme. In this setting, positive gain indicates that BioRARSA outperforms the reverse tracking scheme, and vice versa. By comparing the overall performance, we can see that the insensitivity to the initial choice of parameters allows BioRARSA to achieve better performance on a wide range of initial sampling stepsize compared with the reverse tracking scheme. In certain cases, the performance gain of BioRARSA over

reverse tracking is as large as 29.8% on average.

In section IV it was proven that BioRARSA scales linearly with respect to the system size, and this characteristic is further validated with the simulation shown in Fig. 4. Each line in the figure corresponds to different stopping criteria where the algorithm is stopped when $\mathrm{Mag}(\boldsymbol{\theta}[n]) \geq C_{stop} \cdot \mathrm{Mag}(\boldsymbol{\theta}^\star)$, and $\mathrm{Mag}(\boldsymbol{\theta}^\star)$ is the maximum achievable received signal magnitude as defined in Sec. IV. From this, it is clear that BioRARSA is indeed linearly scalable with target magnitude ranging from 60% to 90% of $\mathrm{Mag}(\boldsymbol{\theta}^\star)$.

In Fig. 5, the evolutions of sampling stepsize Δ_n are presented, where the simulation setup is the same as those in Fig. 1. We observe that the adjusted sampling stepsize first increases drastically and then decreases gradually as the beamforming coefficients approach their optimal values, which confirms with the concept of "optimal relative stepsize[5]" as proposed in [5]. Although only two samples were presented, this phenomenon was observed on most recorded samples.

VI. CONCLUSION

In this work, we propose a bio-inspired distributed beamforming scheme, BioRARSA, based on our understanding of an existing one-bit adaptive scheme via a random search framework, and further inspired by a heuristic algorithm that mimics the bacterial foraging behavior.

Our numerical simulations have validated the proofs derived in Sec. IV, and shown that BioRARSA is more robust and efficient than the one-bit adaptive scheme uniformly and the reverse tracking scheme on average. This improved performance is achieved via the "swim" behavior that adjusts the sampling stepsize adaptively, and hence allowing BioRARSA to be robust against system uncertainties.

In the future, we are interested in further examination of the robustness of BioRARSA under more practical systems. We would also like to improve our framework further such that it can also provide analysis of the rate of convergence.

REFERENCES

[1] R. Mudumbai, B. Wild, U. Madhow, and K. Ramch, "Distributed beamforming using 1 bit feedback: From concept to realization," in *Allerton Conference on Communication, Control, and Computing*, 2006.
[2] C. Lin, V. V. Veeravalli, and S. P. Meyn, "A random search framework for convergence analysis of distributed beamforming with feedback," *IEEE Trans. Information Theory*, vol. 56, no. 12, Dec 2010.
[3] J. Bucklew and W. Sethares, "Convergence of a class of decentralized beamforming algorithms," *IEEE Trans. Signal Processing*, vol. 56, 2008.
[4] C. C. Chen, C. S. Tseng, and C. Lin, "A general proof of convergence for adaptive distributed beamforming schemes," in *International Conference on Acoustics, Speech and Signal Processing*, 2011.
[5] G. Schrack and M. Choit, "Optimized relative step size random searches," *Mathematical Programming*, vol. 10, 1976.
[6] F. J. Solis and R. J.-B. Wets, "Minimization by random search techniques," *Mathematics of Operations Research*, vol. 6, 1981.
[7] K. Passino, "Biomimicry of bacterial foraging for distributed optimization andcontrol," *IEEE Control Systems Magazine*, vol. 22, 2002.

[5]"*Relative stepsize*" is defined as the ratio of the current sampling stepsize to the distance of the current point from the optimal point. In [5], it has been shown that by using optimal relative stepsize in every iteration of a random search algorithm, optimal performance can be achieved.